ELSEVIER

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational and Applied Mathematics 167 (2004) 135–146

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

www.elsevier.com/locate/cam

A global convergent algorithm for flows in a two-dimensional network

H.B. Ji

Department of Automation, University of Science and Technology of China, Hefei, Anhui 230026, China

Received 16 September 2002; received in revised form 30 August 2003

Abstract

In this paper, we will provide an iteration algorithm of global convergence for two-dimensional nonlinear network problems. By avoiding the difficulties of stability of conventional difference methods, our approach is established based on the inner monotone properties of networks, and its global convergence meets the demands of real-time simulation and that of multi-modes of operation. A simulation example is also illustrated for the algorithm.

Keywords: Computer simulation; Thermal networks; Global convergence

1. Introduction

In the computer simulation of industrial processes, a large-scale nonlinear model needs to be solved computationally. The simulation models that we often encounter are stiff systems [4] of ordinary differential equations (ODEs), and a process simulator should be able to work in and switch between many operating modes (startup, shutdown and breakdown modes, etc.), and moreover a real-time simulation is usually required. Because of these requirements, it is usually hard to give a stable difference scheme to solve simulation models. We may fall into a passive position that if one emphasizes on the convergence and precision of simulation then the amount of computation will increase in general and the real-time requirement will be violated.

In this paper we will develop a global convergent algorithm for a class of process models—two-dimensional networks, especially thermal networks [3] with the state variables of pressure and temperature—which commonly exists in many industrial processes. A similar work can be traced back as early to [7], where a global convergent iterative method was proposed for a one-dimensional

E-mail address: jihb@ustc.edu.cn (H.B. Ji).

0377-0427/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2003.09.049

network problem. We will apply some different techniques [2]—the estimation of spectral radius and other simple analytical tools—to prove the global convergence of our method rather than the approach introduced in [7]. The proof is novel to our knowledge. The iteration algorithm in this paper is more general than that of [3] and is easy to practice. In view of the purpose of practical computation, some variations of the method in [2] are derived, including a global convergent Euler's implicit formula with variable step length. A simulation example is illustrated to describe the algorithm. Our procedure can be easily implemented parallelly in modules, which will be useful for realistic process simulation. The methods of thermal networks can also be used for the discretization form of heat-transfer equation (partial differential equation) [5].

2. Nonlinear networks

We consider a nonlinear network consisting of numerous nodes and links. Suppose there are altogether n nodes, and denote the set of nodes by $N = \{1, 2, \ldots, n\}$ and the set of boundary nodes by ∂N. Each node i is determined by two state variables u_i and v_i (that refer to pressure and temperature respectively in thermal networks), and a link lies between two connected nodes. The stream functions ϕ_{ji} and ψ_{ji} represent the rate of mass flow and the rate of energy flow, respectively, from node j to node i over the link (j, i). Assume ϕ_{ji} and ψ_{ji} are functions of the state variables (or potentials) u_i, v_i, u_j, v_j of the two ends of the link (j, i). When node i and j do not connect, let $\phi_{ji} = \psi_{ji} = 0$. The mass stream ϕ_{ji} may have various expressions according to the different situations; the two typical forms are

$$\phi_{ji} = a_{ji} \operatorname{sign}(u_j - u_i)\sqrt{|u_j - u_i|}, \tag{1}$$

$$\phi_{ji} = a_{ji} \operatorname{sign}(u_j^2 - u_i^2)\sqrt{|u_j^2 - u_i^2|}, \tag{2}$$

where

$$\operatorname{sign}(x) = \begin{cases} 1, & x > 0, \\ 0, & x = 0, \\ -1, & x < 0 \end{cases}$$

is the sign function, $a_{ji} = a_{ij} \geqslant 0$ are the coefficients of mass flow. Eq. (1) represents pipe flows of incompressible liquid or of low speed gas; whereas Eq. (2) is similar to the Flügge formula of turbine flow. Sometimes, in the case of gas flow, a_{ji} is related to the "up stream" temperature, i.e., $a_{ji} \propto 1/\sqrt{v_j}$.

The energy stream ψ_{ji} often takes the form

$$\psi_{ji} = b_{ji}(\phi_{ji}^+ + \lambda_{ji})v_j - b_{ij}(\phi_{ij}^+ + \lambda_{ij})v_i, \tag{3}$$

where

$$x^+ = \begin{cases} x, & x \geqslant 0, \\ 0, & x < 0 \end{cases}$$

can be called the positive sign operator, $b_{ji}, b_{ij} > 0$ are the coefficients of energy flow, and $\lambda_{ji}, \lambda_{ij} \geqslant 0$ may be regarded as the constants of heat conduction.

The rate of mass and the rate of energy gained by a node i (total influx) are given respectively by

$$\begin{cases} f_i = \sum_j \phi_{ji}, \\ g_i = \sum_j \psi_{ji} + \theta_i, \end{cases} \tag{4}$$

where $\theta_i = \theta_i(u_i, v_i)$ expresses the effects of chemical reaction or combustion.

The dynamic system of our network problem is

$$\begin{cases} \dfrac{\mathrm{d}u_i}{\mathrm{d}t} = f_i(u,v), \quad \dfrac{\mathrm{d}v_i}{\mathrm{d}t} = g_i(u,v), \ \ i \in N \setminus \partial N, \\ \xi_i(u,v) = 0, \quad \eta_i(u,v) = 0, \ \ i \in \partial N, \\ u_i(0) = u_i^0, \quad v_i(0) = v_i^0, \ \ i \in N, \end{cases} \tag{5}$$

where $u = (u_1, u_2, \ldots, u_n)^{\mathrm{T}}$, $v = (v_1, v_2, \ldots, v_n)^{\mathrm{T}}$. The second row of (5) is the boundary condition, whereas the third row of (5) is the initial condition. For Dirichlet problems, $\xi_i = \bar{u}_i - u_i$, $\eta_i = \bar{v}_i - v_i$, while for Neumann problems, $\xi_i = f_i(u,v) - \bar{f}_i$, $\eta_i = g_i(u,v) - \bar{g}_i$, in which $\bar{u}_i$, $\bar{v}_i$, $\bar{f}_i$ and $\bar{g}_i$ are the prescribed values (the mixed boundary value problem is not included here since it can be constructed by the basic Dirichlet and Neumann problems).

Notice that system (5) usually has stable equilibrium solutions in practical industrial process, we may temporarily avoid the time difference and apply an iteration method to solve the equilibrium points of (5). We hope that the iterative process can somehow reflect the dynamic transition process. With the preliminaries of the iteration method for the associated static system, we will later come to solve the dynamic system (5) using a special time difference method.

The associated equilibrium equations of (5) are

$$\begin{cases} F_i(u,v) = 0, \\ G_i(u,v) = 0, \end{cases} \tag{6}$$

where $i = 1, 2, \ldots, n$ and where we define

$$F_i(u,v) = \begin{cases} f_i(u,v), & i \in N \setminus \partial N, \\ \xi_i(u,v), & i \in \partial N, \end{cases}$$

$$G_i(u,v) = \begin{cases} g_i(u,v), & i \in N \setminus \partial N, \\ \eta_i(u,v), & i \in \partial N. \end{cases}$$

In order to obtain a global convergent algorithm, we require the following hypotheses [2] on stream functions:

(H1) ϕ_{ji} is a continuous function of u_i and u_j.
(H2) ψ_{ji} is a continuous function of u_i, v_i, u_j and v_j; θ_i is a continuous function of u_i and v_i.
(H3) Both ϕ_{ji} and ψ_{ji} are antisymmetric, i.e., $\phi_{ji} = -\phi_{ij}$, $\psi_{ji} = -\psi_{ij}$.
(H4) ϕ_{ji} is monotone increasing in u_j, ψ_{ji} is monotone increasing in v_j.
(H5) The total influx of mass f_i is strictly monotone decreasing in u_i, the total influx of energy g_i is strictly monotone decreasing in v_i.

Note that these hypotheses are not entirely independent and not minimal. For instance, hypothesis (H3) can be weakened [7], but it is not essential for the problem itself. Hypothesis (H1) asserts that the mass stream ϕ_{ji} is independent of the state v, which describes the situation of flows of liquid or that of low-speed gas. From (H3) and (H4) we deduce that ϕ_{ji} is monotone decreasing in u_i and so is ψ_{ji} in v_i. It is easy to verify that the stream functions in (1)–(3) satisfy the whole hypotheses if $\sum_j a_{ji} > 0$ and $\sum_j \lambda_{ij} > 0$. Note that, in the next section we may require the condition of differentiability in the proof of the convergence of algorithm, since the monotone condition of stream functions is given, and by Lebesgue theorem a continuous monotone function is differentiable almost everywhere. To avoid these tedious mathematics [8], we simply use a differential whenever it is required (only in the proof). In fact the following algorithms, when we proceed practically, do not involve any differential.

3. Algorithms

Now we will give a review of the main results of [2], and then we provide some improved algorithms for the sake of practical computation.

The two basic iteration schemes for solving (6) are the iteration of the type of Jacobi and Seidel, which can be expressed as follows:

Jacobi iteration: Solve u_i from the mass equilibrium equation of node i,

$$F_i(u_1^k, \ldots, u_{i-1}^k, u_i, u_{i+1}^k, \ldots, u_n^k) = 0 \Rightarrow u_i = \hat{u}_i^{k+1},$$

adopt the next iterative value of the potential u as

$$u_i^{k+1} = u_i^k + \alpha(\hat{u}_i^{k+1} - u_i^k), \quad 0 < \alpha \leqslant 1,$$

solve v_i from the energy equilibrium equation of node i,

$$G_i(u^k, v_1^k, \ldots, v_{i-1}^k, v_i, v_{i+1}^k, \ldots, v_n^k) = 0 \Rightarrow v_i = \hat{v}_i^{k+1},$$

adopt the next iterative value of the potential v as

$$v_i^{k+1} = v_i^k + \beta(\hat{v}_i^{k+1} - v_i^k), \quad 0 < \beta \leqslant 1,$$

where the node number $i = 1, 2, \ldots, n$, the iteration number $k = 0, 1, \ldots$.

Seidel iteration: Solve u_i from the mass equilibrium equation of node i,

$$F_i(u_1^{k+1},\ldots,u_{i-1}^{k+1},u_i,u_{i+1}^{k},\ldots,u_n^{k})=0 \Rightarrow u_i=\hat{u}_i^{k+1},$$

adopt the next iterative value of the potential u as

$$u_i^{k+1}=u_i^{k}+\alpha(\hat{u}_i^{k+1}-u_i^{k}),\quad 0<\alpha\leqslant 1,$$

solve v_i from the energy equilibrium equation of node i,

$$G_i(u^{k+1},v_1^{k+1},\ldots,v_{i-1}^{k+1},v_i,v_{i+1}^{k},\ldots,v_n^{k})=0 \Rightarrow v_i=\hat{v}_i^{k+1},$$

adopt the next iterative value of the potential v as

$$v_i^{k+1}=v_i^{k}+\beta(\hat{v}_i^{k+1}-v_i^{k}),\quad 0<\beta\leqslant 1,$$

where the node number $i=1,2,\ldots,n$, the iteration number $k=0,1,\ldots$.

Note that though the state u and v are decoupled in Eq. (6) under the preceding hypotheses, we do not compute in advance the solution of $F_i(u)=0$, $i=1,2,\ldots,n$, (numerically compute it until the reach of convergence), and then pass the solution u to the second part $G_i(u,v)=0$, $i=1,2,\ldots,n$, and solve v. Instead, we solve the two subsystems contemporaneously as the iteration methods described above. This is due to the requirements of the real-time simulation.

The above two iteration schemes can be regarded as a point relaxation process, which describes a procedure of computing loops on all nodes of the network—at each node i replacing the potential of node i by a particular value which makes the total influx (mass and energy) vanish at i, while the values of the potential of all the other nodes remain unchanged. α and β can be called relaxation factors.

In order to prove the global convergence of the above iteration, we require a lemma on the estimation of the spectral radius [1].

Lemma 1. *Let $A=(a_{ij})$ be a $n\times n$ irreducible matrix with spectral radius $\rho(A)$, if*

$$\sum_{j=1}^{n}|a_{ij}|\leqslant v,\quad i=1,2,\ldots,n,$$

or

$$\sum_{i=1}^{n}|a_{ij}|\leqslant v,\quad j=1,2,\ldots,n,$$

then $\rho(A) \leqslant v$, *moreover, if there is at least one* k *such that*

$$\sum_{j=1}^{n} |a_{kj}| < v,$$

or

$$\sum_{i=1}^{n} |a_{ik}| < v,$$

then $\rho(A) < v$.

We now provide the main theorem of global convergence.

Theorem 2. *For any finite and connected two-dimensional network that satisfies the conditions from* (H1) *to* (H5), *if there is at least one node at which the Dirichlet boundary condition is given, then the two iterative processes generated by the above Jacobi and Seidel schemes both converge to a unique equilibrium solution from any initial value.*

Proof. We will only sketch the proof of Jacobi iteration; the convergence of Seidel iteration can be inferred from the relation between the two iterative procedures, and a thorough proof can be found in [2].

Denote the sets of Dirichlet boundary nodes by Γ_u and Γ_v (nonempty), which relate to u and v respectively, whereas the residual sets of those are denoted by $\bar{\Gamma}_u = N \setminus \Gamma_u$ and $\bar{\Gamma}_v = N \setminus \Gamma_v$.

From (H5) it follows that the differential of F_i with respect to u_i and that of G_i with respect to v_i are

$$F_{i,i} = \sum_j \phi_{ji,i} < 0, \quad i \in \bar{\Gamma}_u,$$

$$G_{i,i} = \sum_j \psi_{ji,i} + \theta_{i,i} < 0, \quad i \in \bar{\Gamma}_v,$$

and $F_{i,i} = G_{i,i} = -1$ when i is a Dirichlet boundary node. By Implicit Function Theorem, one can solve $u_i = p_i(u)$ and $v_i = q_i(u, v)$ from the equilibrium equations of node i, and thus one can construct the above Jacobi and Seidel iterations.

Now we rewrite the Jacobi scheme in the form of fixed point iteration with relaxation

$$u^{k+1} = \tilde{p}(u^k) = (1 - \alpha)u^k + \alpha p(u^k),$$

$$v^{k+1} = \tilde{q}(u^k, v^k) = (1 - \beta)v^k + \beta q(u^k, v^k),$$

where $p = (p_1, p_2, \ldots, p_n)^{\mathrm{T}}$, $q = (q_1, q_2, \ldots, q_n)^{\mathrm{T}}$, $0 < \alpha, \beta \leqslant 1$, $k = 0, 1, \ldots$, and the related Jacobi matrix is

$$J = \begin{bmatrix} D_u \tilde{p} & 0 \\ D_u \tilde{q} & D_v \tilde{q} \end{bmatrix},$$

where the differentials $D_u \tilde{p} = (1 - \alpha)I + \alpha D_u p$, $D_v \tilde{q} = (1 - \beta)I + \beta D_v q$, and I is the unit matrix.

In order to prove the global convergence of the iteration, it will suffice to show that the spectral radius of J is less than 1 everywhere ($\rho(J) < 1$).

It is easy to verify that $D_u p = (p_{i,j})$ and $D_v q = (q_{i,j})$ have the following expressions:

$$p_{i,j} = \begin{cases} -F_{i,j}/F_{i,i} = \phi_{ji,j} \Big/ \sum_k \phi_{ik,i}, & j \neq i \in \bar{\Gamma}_u, \\ 0, & j = i \text{ or } i \in \Gamma_u, \end{cases}$$

$$q_{i,j} = \begin{cases} -G_{i,j}/G_{i,i} = \psi_{ji,j} \Big/ \left(\sum_k \psi_{ik,i} - \theta_{i,i} \right), & j \neq i \in \bar{\Gamma}_v, \\ 0, & j = i \text{ or } i \in \Gamma_v, \end{cases}$$

in which the antisymmetry of ϕ_{ji} and ψ_{ji} has been considered (H3).

Because of (H4) and (H5) one gets

$$p_{i,j} \geqslant 0, \quad q_{i,j} \geqslant 0, \quad p_{i,i} = q_{i,i} = 0,$$

for all $1 \leqslant i, j \leqslant n$ and this means that both $D_u p$ and $D_v q$ are nonnegative matrices.

Define a similarity transformation on $D_u p$ and on $D_v q$ (without affecting the spectral radius), respectively, as

$$R D_u p R^{-1} = A = (a_{ij}),$$

$$S D_v q S^{-1} = B = (b_{ij}),$$

where

$$R = (r_{ij}) = \operatorname{diag}\left\{ \sum_k \phi_{1k,1}, \sum_k \phi_{2k,2}, \ldots, \sum_k \phi_{nk,n} \right\},$$

$$S = (s_{ij}) = \operatorname{diag}\left\{ \sum_k \psi_{1k,1} - \theta_{1,1}, \sum_k \psi_{2k,2} - \theta_{2,2}, \ldots, \sum_k \psi_{nk,n} - \theta_{n,n} \right\},$$

both A and B remain nonnegative, and their elements are

$$a_{ij} = r_{ii} p_{i,j} r_{jj}^{-1} = \begin{cases} \phi_{ji,j} \Big/ \sum_k \phi_{jk,j}, & j \neq i \in \bar{\Gamma}_u, \\ 0, & j = i \text{ or } i \in \Gamma_u, \end{cases} \tag{7}$$

$$b_{ij} = s_{ii} q_{i,j} s_{jj}^{-1} = \begin{cases} \psi_{ji,j} \Big/ \left(\sum_k \psi_{jk,j} - \theta_{j,j} \right), & j \neq i \in \bar{\Gamma}_v, \\ 0, & j = i \text{ or } i \in \Gamma_v. \end{cases} \tag{8}$$

We can see from (7) that when the node j is not connected with any node in Γ_u, then $\sum_i a_{ij} = 1$, otherwise $\sum_i a_{ij} < 1$. In general, suppose $\Gamma_u = \{i_1, i_2, \ldots, i_t\}$, then the $(n-t) \times (n-t)$ submatrix of A obtained by deleting rows $i_1, \ldots, i_t$ and columns $i_1, \ldots, i_t$ of A is irreducible (since the network is connected), and its $t \times t$ residual submatrix is a zero matrix. It can be verified that the irreducible submatrix satisfies the condition of Lemma 1, hence its spectral radius is less than 1 as asserted. All the eigenvalues of A consist of the eigenvalues of the $(n-t) \times (n-t)$ submatrix and the zero eigenvalues of the $t \times t$ residual submatrix, thus $\rho(D_u p) = \rho(A) < 1$. By analogy, $\rho(D_v q) = \rho(B) < 1$. Since an eigenvalue $\tilde{\lambda}$ of $D_u \tilde{p}$ and an eigenvalue λ of $D_u p$ have the relation $\tilde{\lambda} = 1 - \alpha + \alpha\lambda$, and the spectral radius of a nonnegative matrix is precisely an eigenvalue, thus $\rho(D_u \tilde{p}) \leqslant 1 - \alpha + \alpha\rho(D_u p) < 1$. Furthermore, one can see that $\rho(D_u \tilde{p})$ will decrease in general—which means that the speed of convergence increases [6]—while the factor $\alpha \in (0, 1]$ increases. Similarly one gets $\rho(D_v \tilde{q}) < 1$, therefore $\rho(J) = \max(\rho(D_u \tilde{p}),\ \rho(D_v \tilde{q})) < 1$ everywhere, and the assertion of global convergence readily follows. □

Since the spectral radius of Seidel iteration is not greater than that of Jacobi iteration [2], we come to a conclusion that the speed of Seidel iteration is higher in general than the speed of Jacobi iteration. Moreover, the costs of computer memory in Seidel iteration are lower than that in Jacobi iteration. So the Seidel approach is a more favorite one in practice. From the arguments in the proof of Theorem 2, we may adjust the speed of convergence by changing the relaxation factors α and β, i.e., the bigger $\alpha, \beta \leqslant 1$ are (not greater than 1), the higher the speed of convergence will be. The dynamic transition time that a trainee observes in simulation depends on the speed of convergence. Normally we can set the relaxation factors $\alpha = \beta = 1$, so as to gain the highest speed of convergence. In a training simulator, however, we may change the dynamic transition time or the training difficulties through these relaxation factors to test trainee's operating speed and skills. That is, according to the different levels of a trainee, the simulation process could be slow or fast (super real-time computer simulation).

Note that we may have $F_{i,i} = 0$ or $G_{i,i} = 0$ at a certain node i (due to the closing of valves, for instance). In this situation, the iterative schemes at the node i do not exist any more (the Implicit Function Theorem fails to hold). However, the equilibrium equations are automatically satisfied already at the node, so we can delete the iterative computations of this node. In fact, after such a node is deleted from the whole network, if the network remains connected or the unconnected network consists of several connected subnetworks, each of which has appropriate Dirichlet boundary conditions, then the iterative processes are still convergent.

We can see that many one-variable nonlinear equations are required to be solved in the above iteration procedures, and this—in cases other than the one of solving the energy equilibrium equation (usually linear in v)—is in general quite inconvenient for numerical applications. To this end, by investigating the Seidel iterative process, we can derive the following iteration regulation for the potential u (analogously for v):

Lemma 3. *Under the conditions of Theorem* 2, *if the kth iterative value* u_i^k *at a node i satisfies*

$$F_i(u_1^{k+1}, \ldots, u_{i-1}^{k+1}, u_i^k, u_{i+1}^k, \ldots, u_n^k) \leqslant 0 \ (\geqslant 0),$$

choose the $(k+1)$*th iterative value* u_i^{k+1} (*which can be evaluated by any means*), *such that*

$$u_i^{k+1} \leqslant u_i^k \ (u_i^{k+1} \geqslant u_i^k),$$

$$F_i(u_1^{k+1}, \ldots, u_{i-1}^{k+1}, u_i^{k+1}, u_{i+1}^k, \ldots, u_n^k) \leqslant 0 \ (\geqslant 0),$$

then such an iterative process for u *is still globally convergent*.

Proof. To prove the proposition, we investigate the relation between the iteration of Lemma 3 and the Seidel iteration.

In view of the iterative regulation of Lemma 3 and that of Seidel iteration, we have

$$F_i(u_1^{k+1}, \ldots, u_{i-1}^{k+1}, u_i^{k+1}, u_{i+1}^k, \ldots, u_n^k) \leqslant F_i(u_1^{k+1}, \ldots, u_{i-1}^{k+1}, \hat{u}_i^{k+1}, u_{i+1}^k, \ldots, u_n^k) = 0,$$

which yields $\hat{u}_i^{k+1} \leq u_i^{k+1}$, by the monotone property of the total influx of mass.

Since $u_i^{k+1} \leqslant u_i^k$ as a prerequisite, we obtain $\hat{u}_i^{k+1} \leqslant u_i^{k+1} \leqslant u_i^k$.

Hence there exists an $\alpha \in [0,1]$, such that

$$u_i^{k+1} = u_i^k + \alpha(\hat{u}_i^{k+1} - u_i^k),$$

which implies the new iterative process of Lemma 3 conforms to the original Seidel iteration (except in the trivial case $\alpha = 0$), and it converges accordingly. And the case in parentheses is similar. □

The iteration regulation (Lemma 3) means that in the iterative processes, the sign of the total influx of mass at a node after the current iteration should remain the same as the one before.

Now we propose the following computational method for system (5), its global convergence is obvious by virtue of Lemma 3.

Euler's integration formula with variable step length

1. $\Delta t_u = \Delta t, \ \Delta t_v = \Delta t$
2. $F^0 = F_i(u_1^{k+1}, \ldots, u_{i-1}^{k+1}, u_i^k, u_{i+1}^k, \ldots, u_n^k)$
3. $u_i^{k+1} = u_i^k + F^0 \cdot \Delta t_u$
4. $F^1 = F_i(u_1^{k+1}, \ldots, u_{i-1}^{k+1}, u_i^{k+1}, u_{i+1}^k, \ldots, u_n^k)$
   if F^0 and F^1 have different sign, then go to 5, else go to 6,
5. $\Delta t_u = \frac{F^0}{F^0 - F^1} \cdot \Delta t_u$
   return to 3,
6. $G^0 = G_i(u^{k+1}, v_1^{k+1}, \ldots, v_{i-1}^{k+1}, v_i^k, v_{i+1}^k, \ldots, v_n^k)$
7. $v_i^{k+1} = v_i^k + G^0 \cdot \Delta t_v$
8. $G^1 = G_i(u^{k+1}, v_1^{k+1}, \ldots, v_{i-1}^{k+1}, v_i^{k+1}, v_{i+1}^k, \ldots, v_n^k)$
   if G^0 and G^1 have different sign, then go to 9, else go to 10,
9. $\Delta t_v = \frac{G^0}{G^0 - G^1} \cdot \Delta t_v$
   return to 7,
10. end of this iterative loop,
    where $i = 1, 2, \ldots, n$ and $k = 0, 1, \ldots$.

Here Δt_u and Δt_v stand for the step length associated to u and v respectively, once the initial step length Δt is given, the dynamic step length Δt_u and Δt_v can be automatically adjusted (through a kind of secant method in 5 and 9) to guarantee the global convergence. Note that, since Δt_u and Δt_v may be different, generally an interpolation procedure is needed in order to proceed in the next step of the algorithm, however, our algorithm is simplified, because the convergence and the real-time property are more important than the precision of dynamic transition in the computer simulation of practical plants. The choice of initial step length may depend on the normal time constants of system (5).

In most cases of industrial process simulation, the mass stream ϕ_{ji} is independent (or nearly) of temperature v as hypothesis (H1) shows. Sometimes, however, the variations of temperature may have an effect on mass flow, e.g., the flow formula of a gas turbine (a_{ji} depends on temperature as in (2)). In this case, hypothesis (H1) no longer holds and the convergence of our methods cannot be guaranteed directly. However, because the dynamic transition of pressure is in general much faster than that of temperature, we may adopt a simple revised iteration procedure from a physical point of view like this.

When the residues of mass equilibrium $|F|$ are relatively large (far away from the mass equilibrium point), the effects of the variations of temperature on mass flow can be temporarily neglected (by choosing a very small relaxation factor β or by simply suspending the computation of energy equilibrium equation), whereas, when the residues of mass equilibrium $|F|$ are small, then the coupling temperature and energy equilibrium are taken into account. These treatments of computation usually make the transition process longer, however, it is the necessary cost to ensure the convergence of computation.

4. An example

There are mainly two classes of numerical methods—direct and indirect (iterative) styles [3]—for nonlinear thermal network problems. The direct scheme may converge faster than the iterative one, but it cannot treat thermal networks with nonlinearities other than square root term and its convergence is not global. The iteration algorithm here is an extension of the conventional iteration scheme in that our algorithm can deal with more complicated thermal networks and is easy to practice, though the main ideas are similar.

We now consider a steam network problem shown in Fig. 1. The network consists of 10 nodes and 9 links, where the nodes $\{1,2,3,4\}$ are the interior nodes, and $\{5,6,\ldots,10\}$ are the boundary nodes. The link (1,3) contains a steam turbine, its mass stream is represented by Eq. (2), where $a_{ji} = K/\sqrt{Rv_j}$, in which v_j is the inlet temperature, R is the gas constant, K is the coefficient of flow. All the other links express the normal pipelines, and their mass stream function is described by Eq. (1), where $a_{ji} = A_{ji}\sqrt{2\rho}$, in which ρ is the density of gas, A_{ji} is the cross-area of pipe. The loss of heat is omitted herein (i.e., $\lambda_{ji} = \lambda_{ij} = 0$ in (3)). Node 5 is given a boundary condition of mass flow (Neumann type), and all the rest of the boundary conditions are prescribed to pressure and temperature (Dirichlet type). The boundary conditions are depicted in Table 1.

The steam parameters are: gas constant $R = 0.4625$ kJ/kg/K, specific heat $C_p = 2.289$ kJ/kg/K (e.g. coefficient b_{ji} in (3)), average density $\rho = 4.346$ kg/m^3. The parameters of links (the cross-area of

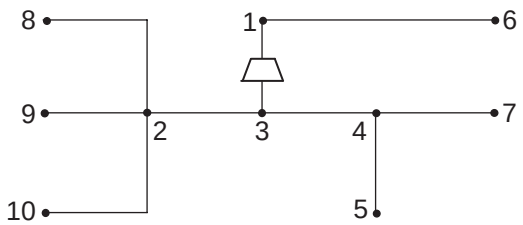

Fig. 1. A network example.

Table 1
Boundary conditions of steam network

Node no.	5	6	7	8	9	10
Pressure/flow	6.0 kg/s	1.0 MPa	0.2 MPa	0.1 MPa	0.5 MPa	0.25 MPa
Temperature (°C)	50	250	100	30	160	90

Table 2
Iterates from initial value

Iteration no.	Node no.	1	2	3	4	5
0	Pressure (MPa)	0.098	0.098	0.098	0.098	0.098
	Temperature (°C)	20.0	20.0	20.0	20.0	50.0
29	Pressure (MPa)	0.9607	0.3391	0.3295	0.2701	0.2359
	Temperature (°C)	250.1	159.3	201.2	201.3	50.0
35	Pressure (MPa)	0.9611	0.3407	0.3313	0.2714	0.2380
	Temperature (°C)	250.0	159.6	197.9	197.9	50.0

pipes or the coefficient of flows) are: $A_{16}=0.0224$, $K_{13}=77.28$, $A_{23}=0.025$, $A_{28}=0.002$, $A_{29}=0.015$, $A_{2,10}=0.009$, $A_{34}=0.016$, $A_{45}=0.011$, $A_{47}=0.007$.

We compute the steam network by using Euler's integration formula with variable step lengths. The initial step length is 0.05 s, the stopping criteria for the algorithm is that the residues of mass equilibrium and that of energy equilibrium at every node are less than 0.08 kg/s and 24 kJ/s, respectively. Table 2 gives selected iterates starting with the initial values of environmental pressure and temperature. The iterates converge to an equilibrium solution in about 35 iterations. After this equilibrium state is arrived at, assume the cross-area of link (2,3) is suddenly reduced by a half (corresponding to the 50% closing of a valve in the pipe), then a new equilibrium state will be reached later. Table 3 illustrates the results under such a throttle, and the iterates converge in about 11 iterations. The computational tests verify the global convergence of the proposed method.

Table 3
Iterates from a sudden throttle

Iteration no.	Node no.	1	2	3	4	5
5	Pressure (MPa)	0.9664	0.3698	0.3574	0.2896	0.2553
	Temperature (°C)	250.0	160.0	201.1	201.5	50.0
11	Pressure (MPa)	0.9671	0.3719	0.3603	0.2924	0.2590
	Temperature (°C)	250.0	160.0	194.5	193.7	50.0

5. Conclusions

We have developed a global convergent computational method for flows in two-dimensional networks, which often exist in the computer simulation of industrial processes. The main computing procedure is a kind of point relaxation process that can be easily implemented parallelly for large-scale computer simulation. The method is suitable for simulation of multi-modes of operation and real-time simulation, and the dynamic transition process can be adjusted by relaxation factors. We have provided an Euler's integration formula with variable step length, and its convergence still holds. We have also discussed the issue of the coupling of pressure and temperature in practical simulation. The method may give some rational directions for the modeling of industrial processes so as to achieve a convergent algorithm.

Acknowledgements

We have benefited from the assistances of computer simulation group in USTC. We also thank the engineers and technicians at Dalian, Qianguo refinery of China for their cooperation.

References

[1] R.A. Horn, C.R. Johnson, Matrix Analysis, Cambridge University Press, Cambridge, 1985.
[2] H.B. Ji, B. Yang, L. Sheng, An extension of an algorithm for flows in a two-dimensional network, J. China Univ. Sci. Tech. 28 (3) (1998) 284–291 (in Chinese).
[3] S.P. Ketkar, Iterative and direct solution methods in thermal network solvers, Internat. Comm. Heat Mass 20 (4) (1993) 527–534.
[4] J. Laffitte, R.M. Howe, Interfacing fast and slow subsystems in the real-time simulation of dynamic systems, Trans. Soc. Comput. Simulation Internat. 14 (3) (1997) 115–126.
[5] M. Milman, W. Petrick, A note on the solution to a common thermal network problem encountered in heat-transfer analysis of spacecraft, Appl. Math. Model. 24 (12) (2000) 861–879.
[6] J. Ortega, W. Rheinboldt, Iterative Solution of Nonlinear Equation in Several Variables, Academic Press, New York, 1970.
[7] T.A. Porsching, Jacobi and Gauss–Seidel methods for nonlinear network problems, SIAM J. Numer. Anal. 6 (3) (1969) 437–449.
[8] W. Rudin, Principles of Mathematical Analysis, McGraw-Hill, New York, 1976.